VIDEO INPAINTING OF OCCLUDING AND OCCLUDED OBJECTS

By

Kedar A. Patwardhan

Guillermo Sapiro

and

Marcelo Bertalmio

IMA Preprint Series # 2016

(January 2005)

INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA
514 Vincent Hall
206 Church Street S.E.
Minneapolis, Minnesota 55455–0436

Phone: 612/624-6066 Fax: 612/626-7370
URL: http://www.ima.umn.edu

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **JAN 2005**	2. REPORT TYPE	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Video Inpainting of Occluding and Occluded Objects**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Office of Naval Research,One Liberty Center,875 North Randolph Street Suite 1425,Arlington,VA,22203-1995**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**5**	

VIDEO INPAINTING OF OCCLUDING AND OCCLUDED OBJECTS

Kedar A. Patwardhan,[§] *Guillermo Sapiro,*[§] *and Marcelo Bertalmio*[¶]

[§]University of Minnesota, Minneapolis, MN 55455, *kedar,guille@ece.umn.edu*
and [¶]Universidad Pompeu-Fabra, Barcelona, Spain

ABSTRACT

We present a basic technique to fill-in missing parts of a video sequence taken from a static camera. Two important cases are considered. The first case is concerned with the removal of non-stationary objects that occlude stationary background. We use a priority based spatio-temporal synthesis scheme for inpainting the stationary background. The second and more difficult case involves filling-in moving objects when they are partially occluded. For this, we propose a priority scheme to first inpaint the occluded moving objects and then fill-in the remaining area with stationary background using the method proposed for the first case. We use as input an optical-flow based mask, which tells if an undamaged pixel is moving or is stationary. The moving object is inpainted by copying patches from undamaged frames, and this copying is independent of the background of the moving object in either frame. This work has applications in a variety of different areas, including video special effects and restoration and enhancement of damaged videos. The examples shown in the paper illustrate these ideas.

1. INTRODUCTION AND OVERVIEW

Videos are an important medium of communication and expression in today's world. Inspite of this, most of the video editing is done manually at the expense of a huge amount of time and money. Hence, the problem of automatic restoration of old movies and automatic object removal and editing for video has begun to attract the attention of many researchers. In this paper we address a constrained but important case of this problem. The constraint being that the camera is fixed, and the scene essentially consists of stationary background with some moving foreground, either of which may require inpainting. The moving object inpainting is background independent.

1.1. Basic Related Work

There has been some preliminary work on frame-by-frame PDEs based video inpainting [1], following [2]. PDE based methods are mainly edge-continuing methods. In [1], the PDE is applied spatially, and completes the video frame-by-frame. This does not take into account the temporal information that a video provides.

The authors in [3] have proposed a method for space-time completion of large damaged areas in a video sequence. They pose the problem of video completion as a global optimization problem with a well-defined objective function, extending to space+time the pioneering static work in [4]. The results shown are for very low resolution videos, and the inpainted static background was different from one frame to another creating a ghost effect. Significant over-smoothing is observed as well.

Very interesting work for repairing damaged video has been recently reported in [5]. Though the results are very impressive in difficult cases, their method involves a gamut of different techniques making the process of inpainting very complicated. A related approach has been reported in [6]. These works combine motion layer estimation and segmentation with warping and region filling-in. We seek a simpler more fundamental approach to the problem of video inpainting.

1.2. Overview of Our Work

Our approach is fundamentally related to the non-parametric sampling method proposed in [4] for the case of 2-D texture synthesis. This method was further improved upon by using a priority and confidence based synthesis in [7]. We have adapted and extended this technique for video inpainting. The general objective of our work is to present a simple and fast method for automatically filling-in videos. The goal of our algorithm is twofold:
(a) To fill-in the static background while maintaining its temporal consistency.
(b) To fill-in the moving foreground while keeping the motion globally consistent.

To achieve this we assume the knowledge (obtained from

This work was partially supported by NSF, ONR, and NGA. We would like to thank Liron Yatziv and Alberto Bartesaghi for interesting discussions. KP would like to thank Arvind Menon for help in preparing test video sequences.

pre-computed optical flow) of whether a pixel is moving or not. This gives us a segmentation of moving foreground and static background. We fill-in the static background by first looking at available temporal information in undamaged frames and "copying" it to damaged frames. This leads to a hole that is common to all the frames which cannot be filled-in temporally. A priority based spatial filling-in scheme is then used to get a best matching patch for the highest priority location. This best match is then copied to all frames. When filling-in moving foreground, we have devised a novel priority scheme that ensures completion of the moving object in each frame. Once the moving object is filled-in, we are left with the simpler case of stationary background filling, which we have already addressed. The subsequent sections describe in more detail our method for video completion and present examples illustrating the ideas.

2. INPAINTING STATIONARY BACKGROUND

In this section we describe our method for filling-in missing stationary background that is occluded by a stationary or moving object.

We first assign confidence values to each pixel in every frame. The confidence of pixels which are deemed to belong to the moving foreground or to the damaged area is set to zero. The rest of the pixels are initialized to a confidence value of one. The process of background filling is completed in two steps:

Temporal Filling-in: We search for the highest priority pixel location in the complete video sequence. This priority computation is similar to [7], and is described in more details later. Temporal information (background pixels) is copied from the temporally nearest undamaged location having the highest confidence.

Spatial Filling-In: Once the temporal filling is over, we are left with a video sequence where all frames have a hole at the same location. We again find the highest priority location to be filled-in, and find a best matching patch. This patch is copied to all the frames, so as to maintain consistent background throughout the sequence.

2.1. Computing Confidence and Filling-In Priority

During the temporal filling-in step, priority of filling-in the 3-D hole is computed in a manner similar to that in [7]. The confidence term $C(p)$, where p is the pixel under consideration, is initialized to zero if p is moving or is damaged and $C(p)$ is initialized to one otherwise. The second relevant term is called the data-term $D(p)$, and its value is based on the availability of temporal information at location p. The data term is computed as follows:

$$D(p) = \frac{\sum_{p \in \partial\Omega, t=-\delta n \ldots \delta n} M_t(p)}{\beta}, \tag{1}$$

where Ω indicates the hole to be filled in , $\partial\Omega$ is its boundary, and $M_t = 0$ if p is damaged or if p is moving, else $M_t = 1$. The time index t indicates the relative position of any frame from the current frame (to which p belongs, $t = 0$). The denominator β is a normalizing constant equal to $2n + 1$, where n indicates the number of previous and next frames considered. Finally the priority of filling-in at $p \in \partial\Omega$ is given by :

$$P(p) = C(p) * D(p) \tag{2}$$

This priority determines the damaged frame and pixel location which we need to first fill-in with background information. We then copy temporal information patches having the highest confidence value from the temporally nearest frame to the location p. Since we are filling-in temporally, and the camera remains fixed, we do not need to perform an explicit search. Such confidence based nearest neighbor copying is better than copying directly from a median image, the median may not contain all the information available in the temporal neighborhood.

Once we copy a patch to the highest priority location p, the confidence at all previously damaged pixels in Ψ_p is updated as in [7][1] :

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I \setminus \Omega)} C(q)}{|\Psi_p|}, \tag{3}$$

where Ψ_p is a patch centered at p, $|\Psi_p|$ is its area, and I denotes the video frame.

When $D(p) = 0, \forall p \in \partial\Omega$, this indicates that there is no more temporal information that can be copied. We then perform a priority based spatial filling-in of the hole, where the data-term in the priority computation follows [7]:

$$D(p) = \frac{|(\nabla \mathbf{I}^{\top})_{\mathbf{p}} \cdot \mathbf{n}_{\mathbf{p}}|}{\alpha}, \tag{4}$$

where I is the grayscale video frame, $\mathbf{n}_{\mathbf{p}}$ is the normal to the hole boundary $\partial\Omega$ at p and α is a normalizing constant (usually 255).

Important to remark here is that the best matching patch found is copied to all the frames, which leads to a consistent background throughout the video sequence. Refer to Figure 3 for an example.

3. INPAINTING MOVING FOREGROUND

We now describe our method for filling-in a moving object that is partially occluded by a stationary or moving second

[1] The $C(p)$ term in Equation (2) is also computed using Equation (3).

object, independent of the changing background from one frame to another. The moving object is filled in frame-by-frame, each frame being completed with the following steps (refer to Figure 1):

1. Find the highest priority location for filling-in the moving object in the current frame (right of Figure 1a).

2. Search for the best matching moving patch from the undamaged portion of the video.

3. Copy only the moving part of the best matching patch to the current frame, and update the confidence values (Equation (3)).

4. Constrain the priority so that any pixel in the copied patch that is not moving gets zero priority (Figure 1c).

5. Repeat until all damaged pixels have zero priority.

The priority computation is similar to that described in the previous section, but the data term is computed differently. In Equation (4), we substitute I by a *motion confidence image* M_c (which we assume is given). $M_c(p)$ is zero if p belongs to the static background, and it is one if p belongs to moving foreground.[2]

The metric used for searching the matching patch is a 5-D SSD, where the 5 elements of the distance vector are the SSD distances between the R, G, B, V_x and V_y values. V_x and V_y are the horizontal and vertical velocity components computed crudely using $V_x = \frac{I_t}{I_x}$ and $V_y = \frac{I_t}{I_y}$, where once again, I is the gray-scale frame, I_t its temporal derivative, and I_x and I_y are its spatial derivatives, all implemented with standard numerical techniques.

Once the moving object is inpainted, the priority for all the remaining pixels turns out to be zero (because of the constraining step 4). The remaining part of the hole can be easily inpainted using the technique described in the previous section.

4. IMPLEMENTATION AND RESULTS

First, Figure 2 completes the moving red rectangle being occluded. This artificial example shows that if we are given a good *motion confidence image* M_c, our method can generate perfectly completed occluded moving objects. In Figure 3 we have successfully removed the person wearing an orange jacket, the phone-box in the center, and the lamp-post towards the left of center. Observe that the inpainted background is consistent throughout the video. Figure 4 shows first the completion of the moving person and then the background. Our algorithm performs very well even when the region to be inpainted is very large. The algorithm was implemented using C++, on a P-4 machine with 768MB of RAM. The complete sequences shown in figures 3 (15 frames) and 4 (70 frames) took less than 5 and 20 mins respectively. All these and more video results can be viewed at *www.tc.umn.edu/*$\sim$*patw*0007*/icip*2005*/index.html*

5. CONCLUDING REMARKS

We presented a simple method for filling-in video sequences. The static background filled-in is consistent throughout the video. Moving object filling-in is independent of changing background from one frame to another and also maintains motion consistency. We assumed that the *motion confidence image* M_c is given and showed that if M_c is really good as in the synthetic case, then the moving object can be inpainted perfectly. We are currently addressing non-binary computations for M_c.

There are a number of issues that we have not addressed, e.g., moving camera. The motion consistency that we achieve can be further improved by using a 3-D search of moving patches as in [7], which would require small patch sizes for the search and also help in filling-in *completely occluded* moving objects. Results in this direction will be reported elsewhere.

6. REFERENCES

[1] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2001.

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," *Computer Graphics (SIGGRAPH 2000)*, 2000.

[3] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," *Proceedings. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004.

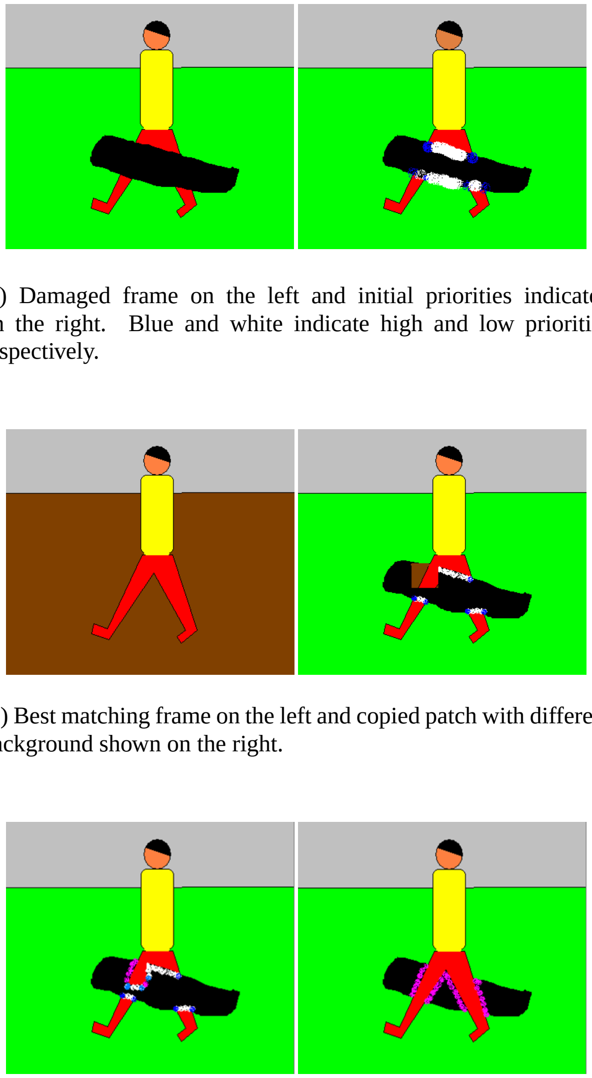
[4] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," *IEEE International Conference on Computer Vision, Corfu, Greece*, 1999.

[5] J. Jia, T. Wu, Y. Tai, and C. Tang, "Video repairing: Inference of foreground and background under severe occlusion," *Proceedings. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004.

[6] Y. Zhang, J. Xiao, and M. Shah, "Motion layer based object removal in videos," *2005 Workshop on Applications of Computer Vision*, 2005.

[7] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based inpainting," *IEEE Transactions on Image Processing*, vol. 9, 2004.

[2] In future work we plan to use optical flow with confidence values to assign non-binary M_c's.

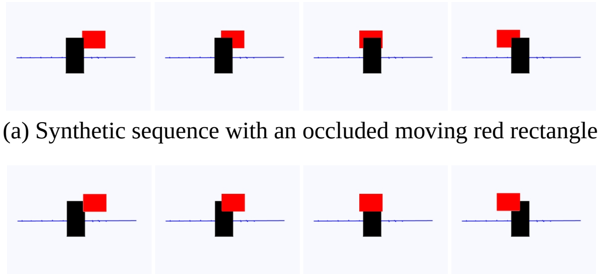


(a) Damaged frame on the left and initial priorities indicated on the right. Blue and white indicate high and low priorities respectively.

(b) Best matching frame on the left and copied patch with different background shown on the right.

(c) Only moving part of the patch is copied (left) and the priority is constrained (purple indicates zero priority). After the moving object is completed there is no damaged pixel with high priority.

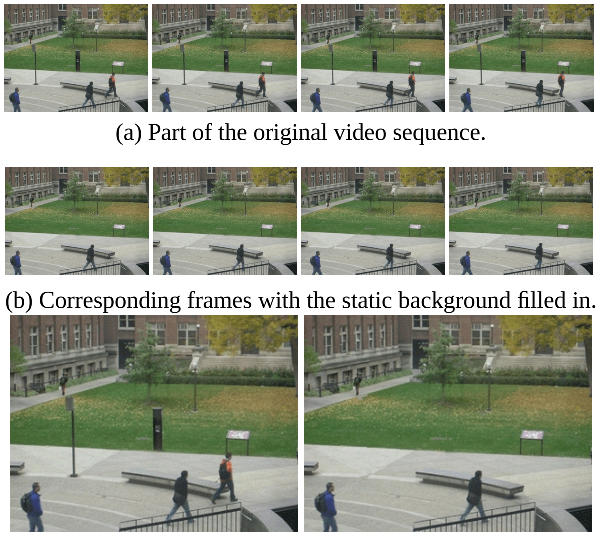
Fig. 1. Overview of the moving object filling-in method.



(a) Synthetic sequence with an occluded moving red rectangle.

(b) Moving rectangle is filled in.

Fig. 2. Synthetic example to show that a perfect *motion confidence image* leads to a perfect moving object completion.

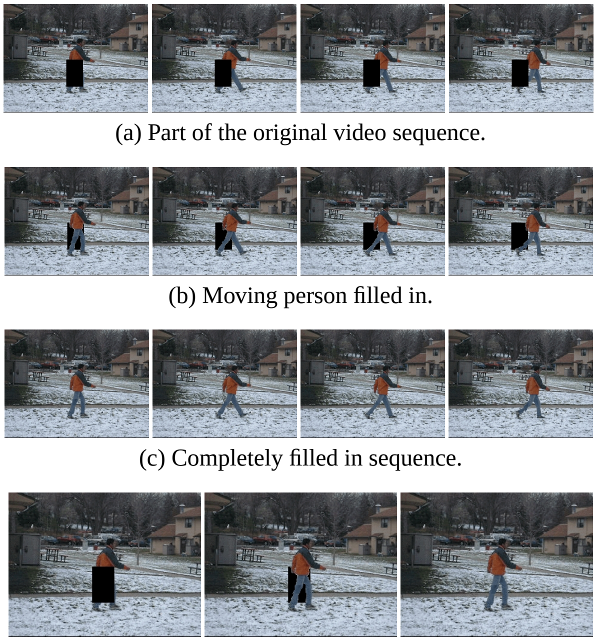


(a) Part of the original video sequence.

(b) Corresponding frames with the static background filled in.

(c) Enlarged results for frame 1 above.

Fig. 3. Static background filling.



(a) Part of the original video sequence.

(b) Moving person filled in.

(c) Completely filled in sequence.

(d) Enlarged results for frame 1 above.

Fig. 4. Moving foreground + static background filling.